Electronic Version

Stylesheet Version v1.1.1

Description

[CONVERTING CIRCUITFORPREVENTING WRONG ERRORCORRECTION CODES FROM OCCURRING DUE TO AN ERROR CORRECTION RULE DURINGDATA READING OPERATION]

BACKGROUND OF INVENTION

[0001] 1. The field of the invention

[0002] The present invention relates to a converting circuit for preventing wrong error correction code from occurring due to an error correction rule during data reading operation. More particularly, when the flash memory controller writes all 0 x FF data into the flash memory, the byte error correction rule generates a set of correct error correction codes and the error correction code converting circuit converts the set of correct error correction codes into 0 x FF error correction codes, and values stored in the data

area and error correction code area of the flash memory are converted into 0 x FF to prevent wrong error correction codes from occurring during data reading operation when the error correction codes are not completely 0 x FF.

[0003] 2. Description of related art

[0004] The flash memory is popular because of its advantageous characteristics, such as low power consumption, non-volatility, shock tolerance and high storage density. The flash memory has gradually replaced EEPROM or battery powered memory in majority portable devices. And the mature semiconductor technology allows further increased the storage density and transmission speed of the flash memory. Thus, the flash memory has successfully replaced the traditional storage media, such as the hard disk.

[0005] For maintaining completeness of the data stored in the flash memory, some manufacturers invented the error correction rule to calculate the value of the error correction code and the error correction rule corrects the bit when the bit error occurs. However, the conventional error correction rule in the flash memory is a single bit error correction rule, wherein the single bit error correction rule can judge whether a single or plurality of damaged bits

exist when the micro processor demands data from the flash memory. If any single bit error occurs, the single bit error correction rule corrects the bit error. On the other hand, when a byte error occurs, the single bit error correction rule will report the error but however the single bit error correction rule cannot correct the byte error. For enabling the single bit error correction rule to correct the byte error, the byte error correction rule is necessary.

[0006] To apply the flash memory into the portable storage device, an erase command must be executed to convert the data area and the error correction code area of the flash memory into 0 x FF, and if the single bit error correction rule is selected, both the data and the error correction codes are converted completely into 0 x FF. Thus, there will be no occurrence of error in error correction codes during the subsequent data reading operation. But when the byte correction rule is selected and when the data is completely 0 x FF and the error correction codes are not completely 0 x FF, then the memory parity error will occur, or even worse, the data reading operation cannot be executed.

[0007] Therefore, it is highly desirable to prevent the wrong error correction codes from occurring due to the byte correc-

tion rule.

Summary of Invention

[0008] Accordingly, in the view of the foregoing, the present inventor makes a detailed study of related art to evaluate and consider, and uses years of accumulated experience in this field, and through several experiments, to create a new converting circuit for allowing increased error bytes tolerance and thereby prevent the error correction code error from occurring due to the byte correction rule.

[0009] According to an aspect of the present invention, the converting circuit is adapted for converting error correction codes calculated by the byte error correction rule into completely 0 x FF when the data is all 0 x FF. Therefore, the occurrence of memory parity error can be effectively reduced or failure of execution data reading operation due to the error correction codes not being completely 0 x FF can be effectively reduced. Accordingly, more byte errors tolerance can be allowed in flash memory by using the byte correction rule.

Brief Description of Drawings

[0010] For a more complete understanding of the present invention, reference will now be made to the following detailed

description of preferred embodiments taken in conjunction with the following accompanying drawings.

[0011] Fig. 1 illustrates a block circuit layout of a flash memory according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0012] Reference will be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers are used in the drawings and the description to refer to the same or like parts.

[0013] Referring to Fig. 1, the flash memory, according to an embodiment of the present invention, comprises a flash memory controller 10, an error correction code area 20 and a data area 30.

[0014] The flash memory controller 10 comprises an error correction code converting circuit 110 having a positive converting circuit 1110 and a negative converting circuit 1120. The error correction code converting circuit 110 is connected to the byte error correction rule 120 connected to a buffer 130.

[0015] The error correction code area 20 is connected to the flash memory controller 10 and the error correction code area 20 is adapted for storing the error correction code

generated by the flash memory controller 10.

[0016] The data area 30 is adapted for storing the data transmitted by the flash memory controller 10 and the data area 30 is connected to the flash memory controller 10.

[0017] When the flash memory controller 10 executes a write command for writing data into the flash memory, the data stored in the buffer 130 is directly written into the data area 30 of the flash memory. Meanwhile, the byte error correction rule 120 generates a set of the error correction codes. The generated error correction codes are converted by the positive converting circuit 1110 of the error correction code converting circuit 110 and the converted error correction codes are stored into the error correction code area 20 of the flash memory.

[0018] When the flash memory controller 10 executes a read command for reading data to the flash memory, the data stored in the data area 30 of the flash memory is directly read into the buffer 130. Meanwhile, the data is transmitted to the byte error correction rule 120. The error correction code of the error correction cord area 20 is reversely converted into the original error correction code by the negative converting circuit 1120 of the error correction code converting circuit 110 and transmitted to the

byte error correction rule 120 in order to calculate and decode the error of the data transmitted back from the data area 30. If the calculated result shows byte error, the error correction can be implemented promptly to correct the byte error in the buffer 130 into the correct value.

[0019] According to the above description, when the flash memory executes an erase command, the value in the data area 30 and the error correction area 20 of the flash memory are all converted into 0 x FF. When the flash memory controller 10 reads data from the flash memory, the data in the error correction code area 20 is converted into the correct error correction code by the error correction code converting circuit 110 into correct the error data. When the flash memory controller 10 writes all 0 x FF data to the flash memory, the byte error correction rule 120 generates a set of the correct error correction codes, which is then converted into the 0 x FF error correction codes by the error correction code converting circuit 110, and finally stored into the data area 30 and the error correction code area 20 of the flash memory as 0 x FF value.

[0020] Without the error correction code converting circuit 110, the data in the buffer 130 is calculated by the byte error correction rule 120 resulting an error correction code,

which is directly stored into the error correction code area 20 of the flash memory. The data read out from the data area 30 of the flash memory is calculated by the byte error correction rule 120 resulting an error correction code and this error correction code is compared with the error correction code read out from the error correction code area 20 of the flash memory. The values of these two error correction codes are equal to each other when there is no error. For example, $E^i f = E^i C$ ($i = 1 \sim N$), wherein:

[0021] $E^i f$: is the error correction code stored in the error correction code area 20

[0022] $E^i C$: is the resulting error correction code

[0023] $\Rightarrow E^1 f, E^2 f, E^3 f \ldots\ldots E^N f = E^1 C, E^2 C, E^3 C \ldots\ldots E^N C$

[0024] With the error correction code converting circuit 110, the error correction code calculated by the byte error correction rule 120 is converted by the positive converting circuit 1110 of the error correction code converting circuit 110 and stored into the error correction code area 20 of the flash memory. The error correction code read out from the error correction code area 20 of the flash memory is reversely converted by the negative converting circuit 1120 of the error correction code converting circuit 110,

and the error correction code is compared with the data read out from the data area 30 of the flash memory and calculated by the byte error correction rule 120, wherein the values of these two are equal each other when there is no error.

[0025] For example, $E^{i}f=E^{i}C \oplus E^{i}A$ (i=1~N), wherein:

[0026] $E^{i}f$: is the error correction code stored in the error correction code area 20

[0027] $E^{i}C$: is the resulting error correction code

[0028] $E^{i}A$: is a constant for the error correction code converting circuit 110

[0029] => $E^{1}f, E^{2}f, E^{3}f\ldots\ldots E^{N}f=(E^{1}C \oplus E^{1}A), (E^{2}C \oplus E^{2}A), (E^{3}C \oplus E^{3}A)\ldots\ldots (E^{N}C \oplus E^{N}A)$

[0030] Wherein, the positive converting circuit 1110 and the negative converting circuit 1120 of the error correction converting circuit 110 are same. For example, $E^{i}C=E^{i}f \oplus E^{i}A$ (i=1~N)

[0031] Assuming values of the data area 30 and the error correction code area 20 of the flash memory are $[D, E^{1}f, E^{2}f, E^{3}f\ldots\ldots E^{N}f]$, when error occurs, the values become $[D \oplus e, E^{1}f \oplus e^{1}f, E^{2}f \oplus e^{2}f, E^{3}f \oplus e^{3}f, \ldots\ldots E^{N}f \oplus e^{N}f]$. Afterwards, the negative converting circuit 1110 of the error correction con-

verting circuit 110 executes the conversion resulting into [$D\oplus e$, $E^1f\oplus e^1f\oplus E^1A$, $E^2f\oplus e^2f\oplus E^2A$, $E^3f\oplus e^3f\oplus E^3A$,.... $E^Nf\oplus e^Nf\oplus E^NA$]= [$D\oplus e$, $E^1C\oplus E^1A\oplus e^1f\oplus E^1A$, $E^2C\oplus E^2A\oplus e^2f\oplus E^2A$, $E^3C\oplus E^3A\oplus e^3f\oplus E^3A$,.... $E^NC\oplus E^NA\oplus e^Nf\oplus E^NA$]= [$D\oplus e$, $E^1C\oplus e^1f$, $E^2C\oplus e^2f$, $E^3C\oplus e^3f$.... $E^NC\oplus e^Nf$].

[0032] According to the above description, although an error correction code converting circuit 110 is added, the address and the value due to actual error are not affected. Therefore, byte error correction rule 120 can function normally and can also solve the problem of data reading with wrong error correction code when the error correction codes are not completely 0 x FF.

[0033] While the invention has been described in conjunction with a specific best mode, it is to be understood that many alternatives, modifications, and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, it is intended to embrace all such alternatives, modifications, and variations in which fall within the spirit and scope of the included claims. All matters set forth herein or shown in the accompanying drawings are to be interpreted in an illustrative and non-limiting sense.